

Learning to Generate Images With Perceptual Similarity Metrics

Karl Ridgeway¹, Jake Snell², Brett D. Roads¹, Richard S. Zemel², and Michael C. Mozer¹

¹ Department of Computer Science, University of Colorado, Boulder

² Department of Computer Science, University of Toronto

Abstract. Deep networks are increasingly being applied to problems involving image synthesis, e.g., generating images from textual descriptions and reconstructing an input image from a compact representation. Supervised training of image-synthesis networks typically uses a pixel-wise loss (PL) to indicate the mismatch between a generated image and its corresponding target image. We propose instead to use a loss function that is better calibrated to human perceptual judgments of image quality: the multiscale structural-similarity score (MS-SSIM) [1]. Because MS-SSIM is differentiable, it is easily incorporated into gradient-descent learning. We compare the consequences of using MS-SSIM versus PL loss on training deterministic and stochastic autoencoders. For three different architectures, we collected human judgments of the quality of image reconstructions. Observers reliably prefer images synthesized by MS-SSIM-optimized models over those synthesized by PL-optimized models, for two distinct PL measures (ℓ_1 and ℓ_2 distances). We also explore the effect of training objective on image encoding and analyze conditions under which perceptually-optimized representations yield better performance on image classification. Just as computer vision has advanced through the use of convolutional architectures that mimic the structure of the mammalian visual system, we argue that significant additional advances can be made in modeling images through the use of training objectives that are well aligned to characteristics of human perception.

1 Introduction

There has been a recent explosion of interest in developing methods for image representation learning, focused in particular on training neural networks to synthesize images. The reason for this surge is threefold. First, the problem of image generation spans a wide range of difficulty, from synthetic images to handwritten digits to naturally cluttered and high-dimensional scenes, the latter of which provides a fertile development and testing ground for generative models. Second, learning good generative models of images involves learning new representations. Such representations are believed to be useful for a variety of machine learning tasks, such as classification or clustering, and can also support transfer between tasks. They are also applicable to other vision problems, including analysis by synthesis, learning of 3D representations, and future prediction in video. Third, image generation is fun and captures popular imagination, as efforts such as Google’s Inceptionism machine demonstrate.

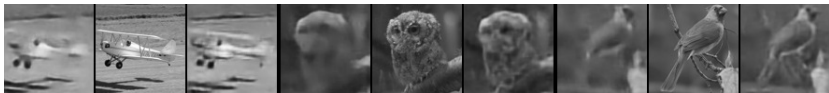


Fig. 1. Three examples showing reconstructions of an original image (center) by a standard reconstruction approach (left) and our technique (right).

While unsupervised image representation learning has become a popular task, there is surprisingly little work on studying loss functions that are appropriate for image generation. A basic method for learning generative image models is the *autoencoder* architecture. Autoencoders are made up of two functions, an encoder and a decoder. The encoder compresses an image into a feature vector, typically of low dimension, and the decoder takes that vector as input and reconstructs the original image as output. The standard loss function is the squared Euclidean distance between the original and reconstructed images; this often yields blurry results. Other methods focus on maximizing the likelihood of generated images, but estimating likelihoods in high-dimensional image space is notoriously difficult [2].

In this paper, we explore the effects of learning representations based on a loss function that, unlike squared reconstruction error and likelihood, is grounded in human perceptual judgements. We show that this perceptually-optimized loss leads to representations that are superior to these other methods, both with respect to reconstructing given images (Figure 1), and generating novel ones. This superiority is demonstrated both in quantitative studies and human judgements. We also examine the classification performance of representations learned via these various loss functions, and analyze conditions when the perceptually-optimized representations are better suited for image classification.

2 Background and Related Work

2.1 Neural networks for image synthesis

The standard neural network for image synthesis is the autoencoder, of which there are two primary types. In the classic *deterministic* autoencoder, the input is mapped directly through hidden layers to output a reconstruction of the original image. The autoencoder is trained to reproduce an image that is similar to the input, where similarity is evaluated using a pixel-wise loss between the image and its reconstruction. In a *probabilistic* autoencoder, the encoder is used to approximate a posterior distribution and the decoder is used to stochastically reconstruct the data from latent variables; the model output is viewed as a distribution over images, and the model is trained to maximize the likelihood of the original image under this distribution. The chief advantage of probabilistic autoencoders is that they permit stochastic generation of novel images. The key issue with probabilistic autoencoders concerns the intractability of inference in the latent variables, e.g., Helmholtz Machines [3]. Variational autoencoders (VAEs) [4] utilize simple variational distributions to address this issue.

A second approach to building generative models for image synthesis uses variants of Boltzmann Machines [5,6] and Deep Belief Networks [7]. While these models are very powerful, each iteration of training requires a computationally costly step of MCMC to approximate derivatives of an intractable partition function (normalization constant), making it difficult to scale them to large datasets.

A third approach to learning generative image models, which we refer to as the *direct-generation* approach, involves training a generator that maps random samples drawn from a uniform distribution through a deep neural network that outputs images, and attempts through training to make the set of images generated by the model indistinguishable from real images. Generative Adversarial Networks (GANs) [8] is a paradigm that involves training a discriminator that attempts to distinguish real from generated images, along with a generator that attempts to trick the discriminator. Recently, this approach has been scaled by training conditional GANs at each level of a Laplacian pyramid of images [9]. With some additional clever training ideas, these adversarial networks have produced very impressive generative results, e.g., [10]. Drawbacks of the GAN include the need to train a second network, a deep and complicated adversary, and the fact that the training of the two networks are inter-dependent and lack a single common objective. An alternative approach, moment-matching networks [11], directly trains the generator to make the statistics of these two distributions match.

Because the goal of image generation is to synthesize images that humans would judge as high quality and natural, current approaches seem inadequate by failing to incorporate measures of human perception. With direct-generation approaches, human judgments could in principle be incorporated by replacing the GAN with human discrimination of real from generated images. However, in practice, the required amount of human effort would make such a scheme unreasonable. In this paper, we describe an alternative approach using the autoencoder architecture; this approach does not require human data collection yet nonetheless incorporates image assessments consistent with human perceptual judgments.

We focus on autoencoders over direct-generation approaches for a second reason: autoencoders *interpret* images in addition to generating images. That is, an input image can be mapped to a compact representation that encodes the underlying properties of the world responsible for the observed image features. This joint training of the encoder and decoder facilitates task transfer: the encoder can be used as the initial image mapping that can be utilized for many different applications. Note, however, that adversarial training is not incompatible with autoencoding, as the two may be combined. In this paper we explore autoencoding alone, to study the effects of optimizing with perceptually-based metrics.

Autoencoders, in contrast to the direct-generation approaches, require the network to reconstruct training images. With such a reconstruction comes the need to evaluate the quality of the reconstruction with respect to the original. Autoencoders are trained based on a pixel-to-pixel comparison of the images—a so-called *full-reference metric*. Deterministic autoencoders typically use mean-squared error (*MSE*), the average square of the pixel intensity differences; they may also use the mean-absolute error (*MAE*), the average of the absolute difference in pixel intensity. Probabilistic autoencoders typically use a likelihood measure that is a monotonically decreasing function of

pixelwise differences. In many instances, these three standard measures—MSE, MAE, and likelihood—fail to capture human judgments of quality. For example, a distorted image created by decreasing the contrast can yield the same standard measure as one created by increasing the contrast, but the two distortions can yield quite different human judgments of visual quality; and distorting an image with salt-and-pepper impulse noise obtains a small perturbation by standard measures but is judged by people as having low visual quality relative to the original image.

2.2 Perception-Based Error Metrics

As digitization of photos and videos became commonplace in the 1990s, the need for digital compression also became apparent. Lossy compression schemes distorted image data, and it was important to quantify the drop in quality resulting from compression in order to optimize the compression scheme. Because compressed digital artifacts are eventually used by humans, researchers attempted to develop full-reference image quality metrics that take into account features to which the human visual system is sensitive and that ignore features to which it is insensitive. Some of these metrics are built on complex models of the human visual system, such as the Sarnoff JND model [12], the visual differences predictor [13], the moving picture quality metric [14], and the perceptual distortion metric [15].

Other metrics take more of an engineering approach, and are based on the extraction and analysis of specific features of an image to which human perception is sensitive. The most popular of these metrics is the structural similarity metric (SSIM) [16], which aims to match the luminance, contrast, and structure information in an image. Other such metrics are the visual information fidelity metric [17], which is an information theory-based measure, and the visual signal-to-noise ratio [18].

Finally, there are transform-based methods, which compare the images after some transformation has been applied. Some of these methods include DCT/wavelets, discrete orthonormal transforms, and singular value decomposition.

2.3 Structural Similarity

In this paper, we train neural nets with the structural-similarity metric (SSIM) [16] and its multiscale extension (MS-SSIM) [1]. We chose the SSIM family of metrics because it is well accepted and frequently utilized in the literature. Further, its pixelwise gradient has a simple analytical form and is inexpensive to compute. In this work, we focus on the original grayscale SSIM and MS-SSIM, although there are interesting variations and improvements such as colorized SSIM [19,20].

The single-scale SSIM metric [16] compares corresponding pixels and their neighborhoods in two images, denoted x and y , with three comparison functions—luminance (I), contrast (C), and structure (S)—defined in Equation 1:

$$I(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad C(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (1)$$

The variables μ_x , μ_y , σ_x , and σ_y denote mean pixel intensity and the standard deviations of pixel intensity in a local image patch centered at either x or y . Following

[16], we chose a square neighborhood of 5 pixels on either side of x or y , resulting in 11×11 patches. The variable σ_{xy} denotes the sample correlation coefficient between corresponding pixels in the patches centered at x and y . The constants C_1 , C_2 , and C_3 are small values added for numerical stability. The three comparison functions are combined to form the SSIM score:

$$\text{SSIM}(x, y) = I(x, y)^\alpha \cdot C(x, y)^\beta \cdot S(x, y)^\gamma \quad (2)$$

This single-scale measure assumes a fixed image sampling density and viewing distance, and may only be appropriate for certain range of image scales. This issue is addressed in [1] with a variant of SSIM that operates at multiple scales simultaneously. The input images x and y are iteratively downsampled by a factor of 2 with a low-pass filter, with scale j denoting the original images downsampled by a factor of 2^{j-1} . The contrast $C(x, y)$ and structure $S(x, y)$ components are applied at all scales. The luminance component is applied only at the coarsest scale, denoted M . Additionally, a weighting is allowed for the contrast and structure components at each scale, leading to the definition:

$$\text{MS-SSIM}(x, y) = I_M(x, y)^{\alpha_M} \cdot \prod_{j=1}^M C_j(x, y)^{\beta_j} \cdot S_j(x, y)^{\gamma_j} \quad (3)$$

In our work, we weight each component and each scale equally ($\alpha = \beta_{1..M} = \gamma_{1..M} = 1$), a common simplification of MS-SSIM. Following [1], we use $M = 5$ downsampling steps.

Our objective is to minimize the loss related to the sum of structural-similarity scores across all image pixels,

$$\mathcal{L}(X, Y) = - \sum_i \text{MS-SSIM}(X_i, Y_i), \quad (4)$$

where X and Y are the original and reconstructed images, and i is an index over image pixels. Equation 4 has a simple analytical derivative, as found in [21], and therefore it is trivial to perform gradient descent in the MS-SSIM-related loss.

We now turn to three sets of simulation experiments that compare autoencoders trained with a pixelwise loss (MSE and MAE) to those trained with a perceptually optimized loss (SSIM or MS-SSIM). The first two experiments are based on deterministic autoencoders, and the third is based on a probabilistic autoencoder, the VAE [4].

3 Deterministic Autoencoders

We demonstrate the benefits of training deterministic autoencoders to optimize SSIM or MS-SSIM across images of various sizes, for various bottlenecks in the autoencoder, and for various network architectures. We begin with a study using small images with a high degree of compression and a fully connected architecture. We then present results on larger images with varying degrees of compression and with a convolutional autoencoder architecture.

3.1 Fully-Connected Architecture

In this simulation, we trained networks on small (32×32) images using a fully-connected architecture in which the first layer expands the 1024-dimensional input to 8192 features, and each successive hidden layer reduces the dimensionality by a factor of two until we reach the bottleneck layer, which has 256 features. The decoder component of the architecture mirrors the encoder. In order to enforce a strong compression of the signal in our autoencoders, we force the activations of units in the bottleneck layer to be binary (-1 or $+1$). Following [22], we threshold the activations in the forward pass and use the original continuous value for the purpose of gradient calculation during back propagation. We perform this quantization both during training and testing and all results reported in this section are based on the quantized bottleneck-layer representations. All layers in the model have ReLU activations, except the bottleneck and output layers which have a tanh activation function. We trained two networks that are identical except for their loss function—one to optimize MSE, and one to optimize SSIM. Because the images are so small, the single-scale SSIM is appropriate; downsampling the images any further blurs the content to the point where humans have trouble distinguishing objects in the image.

Data Sets and Training Methodology. We train autoencoders using a subset of approximately two million images of the 80 million Tiny-Images data set [23], consisting of the first 30 images for every English proper noun. The 32×32 images in this dataset consist of RGB color channels. We mapped the three color channels to a single grayscale channel using the ITU-R 601-2 luma transform. The input pixels are rescaled to the range $[-1, 1]$, to match the tanh activation function on all of our output layers. All testing and evaluation of our models used the CIFAR-10 data set, which consists of 60,000 color images, each drawn from one of ten categories. We chose a diverse data set for training in order ensure that the autoencoders were learning general statistical characteristics of images, and not peculiarities of the CIFAR-10 data set. The CIFAR-10 color images were converted to a single grayscale channel, as was done for the training data set. We divided the CIFAR-10 images into a *search database* (48,000 images) and a *query list* (12,000 images). The purpose of these two subsets will be explained in the classification results section. One use of search database was as a validation set to determine when to stop training: training terminated when the reconstruction error—as measured by the appropriate training metric, either MSE or SSIM—stopped improving following one complete pass through the training set. We train using mini-batches of size 64. The SSIM and MSE metrics are scaled differently, so we performed empirical explorations to set the learning rate appropriately for each. For MSE, we use a learning rate of 5×10^{-5} , and for SSIM 5×10^{-2} . All architectures were trained with a momentum of 0.9 and with weight decay of 5×10^{-5} .

Training Results. As expected, each network performs best on its own training metric: the MSE-optimized network achieves a better reconstruction MSE on 69.71% of the held-out images and the SSIM-optimized network achieves a better reconstruction SSIM score on 97.33% of images.

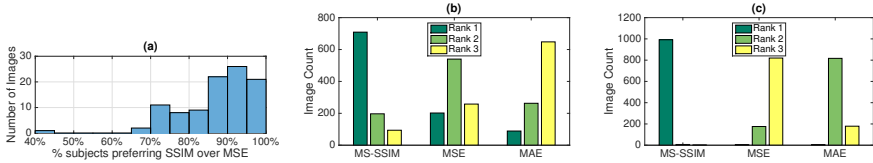


Fig. 2. Human judgments of reconstructed images. (a) Fully connected network: Proportion of participants preferring SSIM to MSE for each of 100 image triplets. (b) Deterministic convolutional network: Distribution of image quality ranking for MS-SSIM, MSE, and MAE for 1000 images from the STL-10 hold-out set. (c) same as (b) for the EL-VAE.

Judgments of Reconstruction Quality. Do human observers prefer reconstructions produced by the SSIM-optimized network or by the MSE-optimized network? We collected judgments of perceptual quality on Amazon Mechanical Turk. Participants were shown image triplets with the original (reference) image in the center and the SSIM- and MSE-optimized reconstructions on either side. Participants were instructed to select which of the two reconstructions they preferred. Half the time the SSIM-optimized reconstruction appeared on the left and half the time it appeared on the right.

In a first study, twenty participants provided preference judgments on the same set of 100 randomly selected images from the CIFAR-10 data set. For each image triple, we recorded the proportion of participants who choose the SSIM reconstruction of the image over the MSE reconstruction. Figure 2a shows the distribution of inter-participant preference for SSIM reconstructions across all 100 images. If participants were choosing randomly, we would expect to see roughly 50% preference for most images. However, a plurality of images have over 90% inter-participant agreement on SSIM, and almost no images have MSE reconstructions that are preferred over SSIM reconstructions by a majority of participants.

Figure 3a shows the eight image triplets for which the largest proportion of participants preferred the SSIM reconstruction. The original image is shown in the center of the triplet and the MSE- and SSIM-optimized reconstructions appear on the left and right, respectively. (In the actual experiment, the two reconstructions were flipped on half of the trials.) The SSIM reconstructions all show important object details that are lost in the MSE reconstructions and were unanimously preferred by participants. Figure 3b shows the eight image triples for which the smallest proportion of participants preferred the SSIM reconstruction. In the first seven of these images, still a majority (70%) of participants preferred the SSIM reconstruction to the MSE reconstruction; only in the image in the lower right corner did a majority prefer the MSE reconstruction (60%). The SSIM-optimized reconstructions still seem to show as much detail as the MSE-optimized reconstructions, and the inconsistency in the ratings may indicate that the two reconstructions are of about equal quality.

In a second study on Mechanical Turk, twenty new participants each provided preference judgments on a randomly drawn set of 100 images and their reconstructions. The images were different for each participant; consequently, a total of 2000 images were judged. Participants preferred the SSIM- over MSE-optimized reconstructions by

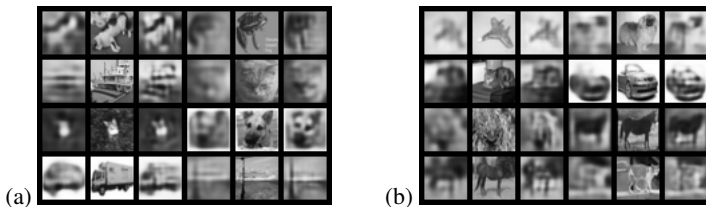


Fig. 3. Image triples consisting of—from left to right—the MSE reconstruction, the original image, and the SSIM reconstruction. Image triples are ordered, from top to bottom and left to right, by the percentage of participants preferring SSIM. (a) Eight images for which participants strongly preferred the SSIM reconstruction over the MSE reconstruction. (b) Eight images for which the smallest proportion of participants preferred the SSIM reconstruction.

nearly a 7:1 ratio: the SSIM reconstruction was chosen for 86.25% of the images. Examining individual participants, The participant choosing SSIM reconstructions the least still preferred them 63% of the time, and the participant choosing SSIM reconstructions the most preferred them 99% of the time.

3.2 Convolutional Architecture

Next we trained networks on larger images (96×96 pixels). For these simulation, we use a *convolutional* autoencoder architecture [24]: convolutional layers encode the input and deconvolutional layers decode the feature representation in the bottleneck layer. The deconvolutional layers are implemented as convolutional layers that are preceded by an upsampling step that creates a layer with 2 times the dimensions of the input layer by repeating the values of the input. To explore the role of the capacity of the convolutional layer, we built models with bottlenecks of both 128 and 512 units. However, we report only results for the 128-unit network as qualitative performance of the 512-unit network was quite similar.

The convolutional network architecture consists of 3 convolutional layers, each with a filter size of 5 and a stride of 2, which means the input is reduced from (96,96) to (48,48), then to (24,24), and finally to (12,12). The first layer has 128 filters, the second 256, and the final 512. All layers have ReLU activations, except the bottleneck and output, which are tanh. The deconvolutional layers mirror the convolutional layers.

For these larger images, which may have structure at multiple spatial scales, we used the MS-SSIM rather than SSIM as our perceptual similarity metric. We compared MS-SSIM to two pixelwise measures: MSE and MAE. Because MSE focuses on outliers and we have no reason to believe that the human eye has a similar focus, we felt it important to include MAE. If we observe MS-SSIM outperforming both MSE and MAE, we will have stronger evidence for the conclusion that perceptually-optimized measures outperform pixelwise losses in general.

Data Sets and Training Methodology. For training and testing, we use the STL-10 dataset [25], which consists of larger RGB color images, of the same classes as CIFAR-10. The images were converted to grayscale using the method we used for CIFAR-10.

For our experiments, we train our models on the 100,000 images in STL-10 referred to as the ‘unlabeled’ set, and of the remaining data, we formed a *validation* set of 10,400 images and a *test* set of 2,800 images. The validation set is used to determine when to stop training. Other than early stopping, no regularization was used during training. We used the “Adam” optimizer [26].

Training Results. Table 1 shows the held-out reconstruction performance for each model, on each loss function. As expected, in each case the model trained on a particular loss will do best on that loss, indicating that the models have all trained properly.

Judgments of Reconstruction Quality. We performed a Mechanical Turk study to determine whether human observers prefer images generated by the MS-SSIM-optimized networks to MSE- and MAE-optimized networks. Images were chosen randomly from the validation set. Participants were presented with a sequence of screens showing the original (reference) image on the left and a set of three reconstructions on the right. Participants were instructed to drag and drop the images vertically into the correct order, so that the best reconstruction is on top and the worst on the bottom. The initial vertical ordering of reconstructions was randomized. We asked 20 participants to each rank 50 images, for a total of 1000 rankings. Figure 2b shows the distribution over rankings for each of the three training objectives. If participants chose randomly, one would expect to see the same number of high rankings for each model. However, MS-SSIM is ranked highest for a majority of images (709 out of 1000).

Figure 4a shows four examples of images whose MS-SSIM reconstruction was ranked by human judges as the best. Figure 4b shows four examples of images whose MSE or MAE reconstruction was ranked as the best. The strong preference for MS-SSIM appears to be due to its superiority in capturing fine detail such as the monkey and cat faces and background detail such as the construction cranes. MS-SSIM seems to have less of an advantage on simpler, more homogeneous, less textured images. Note that even when MSE or MAE beats MS-SSIM, the MS-SSIM reconstructions have no obvious defects relative to the other reconstructions.

4 Probabilistic Autoencoders

In order to further explore the role of perceptual losses in learning models for image generation, we adapt the VAE model of [4] to be trained with an arbitrary differentiable image similarity metric. The VAE model closely resembles a standard autoencoder, utilizing a combination of an encoding network that produces a code for an image x ,

| Train Loss | MSE | MAE | SSIM | MS-SSIM |
|------------|---------------|---------------|---------------|---------------|
| MSE | 0.0233 | 0.1012 | 0.6616 | 0.5212 |
| MAE | 0.0262 | 0.0964 | 0.6667 | 0.5178 |
| MS-SSIM | 0.0271 | 0.1064 | 0.7134 | 0.6018 |

Table 1. Reconstruction Losses for Deterministic Convolutional Autoencoder

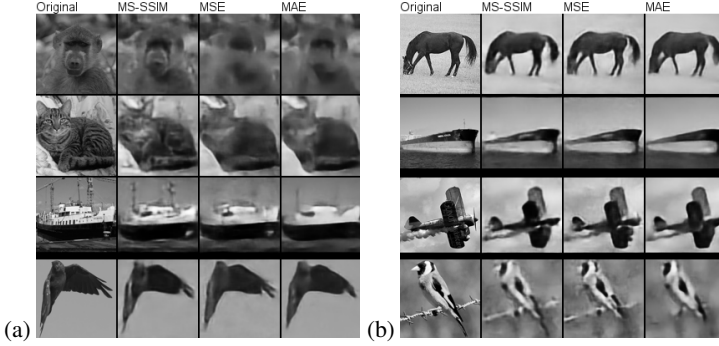


Fig. 4. (a) Four randomly selected, held-out STL-10 images and their reconstructions for the 128-hidden-unit networks. For these images, the MS-SSIM reconstruction was ranked as best by humans. (b) Four randomly selected test images where the MS-SSIM reconstruction was ranked second or third.

and then a decoding network that maps the code to an image \hat{x} . The key difference is that the code z is considered a latent variable, with a prior $p(z)$. The encoder $q_\phi(z|x)$, parameterized by ϕ , approximates the intractable posterior of z given the image x . The decoder $p_\theta(x|z)$, parameterized by θ , produces a distribution over images given z . The VAE minimizes a variational upper bound on the negative log-likelihood of the data:

$$-\log p_\theta(x^{(n)}) \leq -\mathbb{E}_{q_\phi(z|x^{(n)})} [\log p_\theta(x^{(n)}|z)] + D_{KL}(q_\phi(z|x^{(n)})||p(z)) \quad (5)$$

$$= \mathcal{L}^{VAE}(\phi, \theta, x^{(n)}) \quad (6)$$

We modify this learning objective to better suit an arbitrary differential loss $\Delta(x, \hat{x})$ by replacing the probabilistic decoder with a deterministic prediction as a function of the code: $\hat{x} \equiv f_\theta(z)$. The objective then becomes a weighted sum of the expected loss of \hat{x} under the encoder’s distribution over z and the KL regularization term:

$$\mathcal{L}^{EL}(\phi, \theta, x^{(n)}) = C \cdot \mathbb{E}_{q_\phi(z|x^{(n)})} [\Delta(x, \hat{x})] + D_{KL}(q_\phi(z|x^{(n)})||p(z)), \quad (7)$$

where the constant C governs the trade-off between the image-specific loss and the regularizer. We call this modification *Expected-Loss VAE* (EL-VAE).

Experimental Details of EL-VAE Training. We trained convolutional EL-VAE networks on 96×96 pixel images with a similar architecture to the deterministic convolutional autoencoders. For the encoder, we again used 3 convolutional layers, again with a filter size of 5 and a stride of 2. The number of filters matched the deterministic case: 128 in the first layer, 256 in the second layer, 512 in the final convolutional layer, and a bottleneck of size 128. We used a Gaussian MLP [4] to form the approximate posterior distribution. We chose $\mathcal{N}(0, \mathbf{I})$ as the prior. The deconvolutional layers again mirrored the convolutional layers and we used tanh units at the output.

EL-VAE Training Methodology. We trained convolutional EL-VAEs on the unlabeled portion of STL-10 until performance on the STL-10 training set no longer improved. No regularization was used during EL-VAE training, and the “Adam” optimizer [26] was again used.

The value of C in the EL-VAE objective (Equation 7) governs the trade-off between the KL loss and reconstruction error. As C increases, the model will put greater emphasis on reconstructions. At the same time, the KL-divergence of the prior from the approximate posterior will increase, leading to poorer samples. Selecting a value of C is further complicated due to the different scaling depending on the choice of the image-specific loss Δ .

In order to mitigate the differences in scaling, we normalized each loss (MSE, MAE, and negative MS-SSIM) by dividing by its expected value as estimated by computing the loss on 10,000 pairs randomly drawn with replacement from the training set. To select the best value of C , we utilized a recent approach to model selection in generative models [27]. This work proposes a statistical test of relative similarity to determine which model generates samples that are significantly closer to the reference dataset of interest. The test statistic is the difference in squared maximum mean discrepancies (MMDs) between the reference dataset and a dataset generated by each model. We trained convolutional EL-VAEs with $C \in \{1, 10, 1000, 10000\}$ for each loss on a 5,000 example subset of the STL-10 unlabeled dataset. We then utilized the test statistic of [27] to determine for each loss the value of C that produced samples with smallest squared MMD compared to the STL-10 train set. We used a sample size of 800, both for the reference dataset and the model sample sets. For MSE and negative MS-SSIM, the best value was $C = 1000$. For MAE, the test was inconclusive 7 times out of 10 when deciding between $C = 100$ and $C = 1000$, so for the sake of simplicity we chose $C = 1000$. Thus $C = 1000$ was used when training EL-VAEs on the full unlabeled STL-10 dataset.

Judgments of Reconstruction Quality. We performed a final Mechanical Turk study to determine human observer preferences for image reconstructions generated by MS-SSIM-, MSE-, and MAE-optimized EL-VAE architecture. To do so, we generated reconstructions of 1000 randomly chosen images from the STL-10 test set by taking the latent code to be the mode of the approximate posterior for each EL-VAE network. The study procedure was the same as detailed above for the reconstructions of 96×96 images in the deterministic case. MS-SSIM was ranked the highest in 992 out of 1000 cases. Figure 2c shows the distribution of image rankings for each loss. Test reconstructions for the EL-VAE networks are shown in Figure 5. Figure 5a shows reconstructions for which MS-SSIM was ranked as best, and Figure 5b shows reconstructions for which it was not ranked best. As observed in the deterministic case, MS-SSIM is better at capturing fine details than either MSE or MAE.

Qualitative Evaluation of Generated Samples. In order to qualitatively assess the performance of each EL-VAEs as a generative model, Figure 6 shows random samples from each model. Each image was generated by drawing a code z from the prior

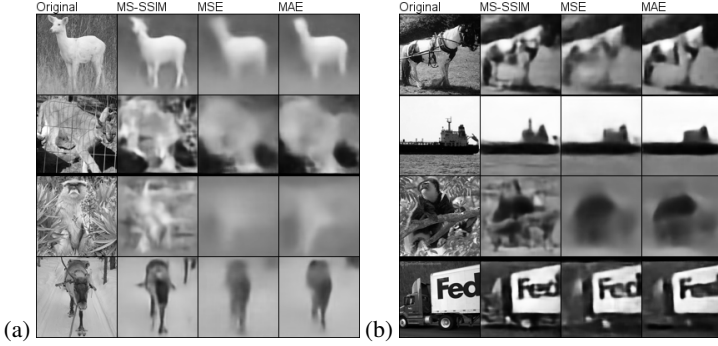


Fig. 5. (a) Four randomly selected, held-out STL-10 images and their reconstructions. For these images, the MS-SSIM reconstruction was ranked as best by humans. Reconstructions are from the 128-hidden-unit VAEs. From left to right are the original image, followed by the MS-SSIM, MSE, and MAE reconstructions. (b) Four randomly selected test images where the MS-SSIM reconstruction was ranked second or third.

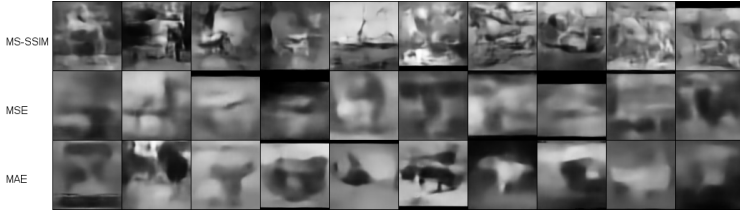


Fig. 6. Samples generated from EL-VAEs optimized with MS-SSIM (top row), MSE (middle row) and MAE (bottom row). The quality of samples appears to mirror those of the reconstructions, as MS-SSIM optimized EL-VAE is able to generate fine details that the other models do not.

and then passing it as input to the decoder. The samples generated by the MS-SSIM-optimized net contain a great degree of detail and structure.

5 Using Learned Representations for Classification

In the previous sections, we showed that using a perceptually-aligned training objective improves the quality of image synthesis, as judged by human observers, for three different neural net architectures. In this section, we investigate whether the SSIM objective leads to the discovery of internal representations in the neural net that are more closely tied to the high-level object category associated with an image—namely, the 10 object categories in CIFAR-10/STL-10. One hypothesis is that, to the extent that image reconstructions look better, image classification should improve because critical features of the images are preserved. An alternative hypothesis is that classification involving a small number of alternatives requires only low-spatial-frequency information,

and faithful reconstructions of object details and background textures (e.g. what we see in the SSIM-based reconstructions) might not be helpful.

We examine the compressed representations of the image in the bottleneck layer, which we refer to as the image *code*. We use these codes to perform three image classification experiments—one using a k -nearest-neighbor search on the 256-bit binary codes we learned for the CIFAR-style images, and two using support-vector machine classifiers on the 128-hidden-unit codes discovered by the deterministic convolutional architecture and the EL-VAE on the STL-10 dataset.

5.1 Fully-Connected Autoencoder Trained on CIFAR-10

If the SSIM objective biases learning toward the discovery of codes that convey useful information about the object present in an image, then we should see categorical clustering of codes. That is, the code associated with the image of one dog should be more similar to codes for images of other dogs than perhaps the code for an image of a visually similar cat. Using the method of [22], we probe the network with a set of *query* images and we use the corresponding code to index into a *search database*—a set of 48,000 images whose codes and category labels have been stored. Using the search database to identify the k nearest neighbors in Hamming distance, we can compute the proportion of the nearest k that have the same category label as the query image. Codes that embody category information will yield a higher score.

The results for the MSE and SSIM models are very close. Averaging the accuracy of the k -NN classifiers for $k \in \{1, \dots, 10\}$, the MSE code classification is 39.37% accurate, and the SSIM code classification is 39.78% accurate. Thus, the SSIM-optimized codes do not seem to be any more useful for k -NN classification than the MSE-optimized codes.

5.2 Convolutional Autoencoder Trained on STL-10

We performed another classification experiment on the continuous valued, 128-hidden-unit codes learned on the STL-10 dataset. For each training objective (MSE, MAE, MS-SSIM), we fit a radial-basis-function kernel support vector machine (RBF-SVM) to the codes obtained for inputs in the validation set we created. Hyperparameters of the RBF-SVM were set using cross-validation on the validation set. We then used the RBF-SVM to classify images in the test set, and computed mean classification accuracy. Again, the results between models are fairly close: MS-SSIM training achieves an accuracy of 46%, MAE 48.5%, and MSE 45.6%. These differences are likely meaningless.

5.3 Variational Autoencoder Trained on STL-10

We performed a final classification experiment on representations extracted from each EL-VAE model trained on STL-10. We generated a code for each example of the STL-10 train set as the mode of the approximate posterior conditioned on the image. We again used an RBF-SVM, this time fitted on the codes of the STL-10 train set, and cross-validated against the STL-10 test set to find the optimal value of the SVM hyperparameters. MS-SSIM achieved an accuracy of 45.1%, followed by MAE at 44.9% and MSE at 41.8%.

6 Discussion and Future Work

We have investigated the consequences of replacing pixel-wise loss functions, MSE and MAE, with perceptually-grounded loss functions, SSIM and MS-SSIM, in neural networks that generate images. Human observers judge SSIM-optimized images to be of higher quality than PL-optimized images over a range of neural network architectures.

For image classification, we failed to find an advantage for codes learned using SSIM or MS-SSIM optimized networks. This supports our hypothesis that the differences in image content which make MS-SSIM reconstructions appear better tend to relate to attributes of the image only weakly related to the object class, such as background details and fine-grained textures. While we may not have shown an improvement on the CIFAR classification task, the improved detail in reconstructions indicates that MS-SSIM trained codes might be more useful for fine-grained classification tasks, such as bird species identification.

With respect to the experiments on stochastic image generation, we plan to go beyond qualitative assessment of the EL-VAE model by evaluating held-out data likelihood. This could be accomplished for example by Parzen window estimation, or by Hybrid Monte Carlo (HMC) methods as done by [4]. We are also investigating how probabilistic generative models can be formulated by taking into account perceptual loss.

A recent manuscript [28] also proposed using SSIM and MS-SSIM as a training objective for image processing neural networks. In this manuscript, the authors evaluate alternative training objectives based not on human judgments, but on a range of image quality metrics. They find that MAE outperforms MSE, SSIM, and MS-SSIM on their collection of metrics, and not surprisingly, that a loss which combines both PL and SSIM measures does best—on the collection of metrics which include PL and SSIM measures. Our work goes further in demonstrating that perceptually-grounded losses attain better scores on the definitive assessment of image quality: that registered by the human visual cortex.

Given our encouraging results, it seems appropriate to investigate other perceptually-grounded loss functions. SSIM is the low-hanging fruit because it is differentiable. Nonetheless, even black-box loss functions can be cached into a *forward model* neural net [29] that maps image pairs into a quality measure. We can then back propagate through the forward model to transform a loss derivative expressed in perceptual quality into a loss derivative expressed in terms of individual output unit activities. This flexible framework will allow us to combine multiple perceptually-grounded loss functions. Further, we can refine any perceptually-grounded loss functions with additional data obtained from human preference judgments, such as those we collected in the present set of experiments.

Acknowledgements

The first two authors contributed equally to this work. This research was supported by NSF grants SES-1461535, SBE-0542013, and SMA-1041755.

References

1. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multi-scale structural similarity for image quality assessment. *IEEE Asilomar Conference on Signals, Systems and Computers* **2** (2003) 9–13
2. Theis, L., van den Oord, A., Bethge, M.: A note on the evaluation of generative models. In: *International Conference on Learning Representations*. (2016)
3. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The Helmholtz machine. *Neural Computation* **7**(5) (1995) 889–904
4. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *International Conference on Learning Representations – ICLR*. (2014)
5. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D.E., McClelland, J.L., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1. MIT Press, Cambridge, MA (1986) 194–281
6. Hinton, G.E., Sejnowski, T.J.: Learning and relearning in boltzmann machines. In Rumelhart, D.E., McClelland, J.L., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1: Foundations. Volume 1. MIT Press, Cambridge, MA (1986) 283–317
7. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. *Neural Computation* **18**(7) (2006) 1527–54
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M.: Generative adversarial networks. *arXiv 1406.266v1 [stat.ML]* (2014) 1–9
9. Denton, E., Chintala, S., Szlam, A., Fergus, R.: Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *arXiv 1506.05751 [stat.ML]* (2015) 1–10
10. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015)
11. Li, Y., Swersky, K., Zemel, R.: Generative Moment Matching Networks. In: *Proceedings of The 32nd International Conference on Machine Learning*. (2015) 1718–1727
12. Lubin, J.: A human vision system model for objective image fidelity and target detectability measurements. In: *Proc. EUSIPCO*. Volume 98. (1998) 1069–1072
13. Daly, S.J.: Visible differences predictor: an algorithm for the assessment of image fidelity. In: *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, International Society for Optics and Photonics (1992) 2–15
14. Van den Branden Lambrecht, C.J., Verscheure, O.: Perceptual quality measure using a spatiotemporal model of the human visual system. In: *Electronic Imaging: Science & Technology*, International Society for Optics and Photonics (1996) 450–461
15. Winkler, S.: A perceptual distortion metric for digital color images. In: *ICIP* (3). (1998) 399–403
16. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4) (2004) 600–612
17. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. *IEEE Transactions on Image Processing* **15**(2) (2006) 430–444
18. Chandler, D.M., Hemami, S.S.: Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing* **16**(9) (2007) 2284–2298
19. Kolaman, A., Yadid-Pecht, O.: Quaternion structural similarity: a new quality index for color images. *Image Processing, IEEE Transactions on* **21**(4) (2012) 1526–1536
20. Hassan, M., Bhagvati, C.: Structural Similarity Measure for Color Images. *International Journal of Computer Applications* (0975 8887) **43**(14) (2012) 7–12
21. Wang, Z., Simoncelli, E.P.: Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities. *Journal of Vision* **8**(12) (2008) 8

22. Krizhevsky, A., Hinton, G.E.: Using very deep autoencoders for content-based image retrieval. In: ESANN, Citeseer (2011)
23. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for non-parametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **30**(11) (2008) 1958–1970
24. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: *Artificial Neural Networks and Machine Learning–ICANN 2011*. Springer (2011) 52–59
25. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: *International conference on artificial intelligence and statistics*. (2011) 215–223
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
27. Bounliphone, W., Belilovsky, E., Blaschko, M.B., Antonoglou, I., Gretton, A.: A test of relative similarity for model selection in generative models. *arXiv preprint arXiv:1511.04581* (2016)
28. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Is l2 a good loss function for neural networks for image processing? *arXiv preprint arXiv:1511.08861* (2015)
29. Jordan, M.I., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. *Cognitive Science* **16**(3) (1992) 307–354